

# Einführung in die computerlinguistische Programmierung mit Python

## 04-02: Dateien schreiben

Der Inhalt von Variablen ist immer bis zum Ende der Laufzeit verfügbar. Ausgaben, die wir mit `print()` umsetzen, sind nur so lange sichtbar, bis wir das Terminal schließen. Wenn wir Informationen **langfristig speichern** wollen, können wir Dateien im Dateisystem des Computers anlegen und sie mit Inhalten füllen.

Der folgende Code erzeugt eine Datei und schreibt einige Zeilen in diese Datei:

```
In [ ]: filename = "secrets.txt"

# Mit dieser Zeile öffnen wir die Datei
# zum Schreiben ("w" wie "write").
with open(filename, "w") as outfile:

    # Hier definieren wir den Inhalt,
    # der in die Datei geschrieben
    # werden soll.
    content = """Ich mag Python mehr als Java.
Ich mag Kuhmilch mehr als Sojamilch.
Ich mag Herbst lieber als Frühling."""

    # Dieser print-Befehl gibt den Inhalt
    # in die Datei outfile aus.
    print(content, file=outfile)

    # Dieser print-Befehl gibt Informationen direkt hier aus.
    print("Daten wurden in der Datei " + filename + " gespeichert!")
```

Den Dateinamen können wir frei wählen. Wir verwenden meist die Dateiendung `.txt` für Textdateien. Wenn wir nur den Namen angeben, wird die Datei im gleichen Verzeichnis erzeugt, in dem unser Pythoncode ausgeführt wird. Um einen anderen Speicherort zu wählen, geben wir absolute Pfade an, z.B. `filename = "D:/Files/secrets.txt"`.

```
In [ ]: #####
# OPTIONAL/HINWEIS #
# Die folgenden zwei Zeilen könnt ihr an #
# den Anfang eures Codes kopieren, um herauszufinden, #
# in welchem Ordner der Code ausgeführt wird (und #
# wo demnach die neue Datei gespeichert wird): #
#####

import os
print(os.getcwd())
```

Die Zeile, die mit `with` beginnt, öffnet die Datei. Danach folgende Zeilen sind eingerückt: Nur innerhalb des Blocks können wir auf das Objekt `outfile` zugreifen.

Die allgemeine Form des Codes zum Öffnen einer Datei im Schreibmodus sieht so aus:

```
with open(<dateiname>, "w") as <dateivARIABLE>:
    ...
    print(<inhalt>, file=<dateivARIABLE>)
```

Für alle eingerückten Zeilen ist die geöffnete Datei verfügbar. Sobald wieder Code ohne Einrückung folgt, wird die Datei automatisch vom Interpreter **geschlossen**.

Die DateivARIABLE ist ein Name, den wir frei wählen können, ähnlich wie bei der Laufvariable in `for`-Schleifen. Diese Variable brauchen wir, um Inhalte in die Datei zu schreiben: Wir verwenden hierfür `print` mit dem zusätzlichen Parameter `file`. Dabei muss der Wert für `file` der Variablenname sein, den wir beim Öffnen der Datei erzeugt haben.

Wir haben oben den Modus `"w"` **wie** `write` (schreiben) gewählt. Bei jeder Ausführung des Codes wird die Datei **komplett überschrieben**. Wenn wir stattdessen `"a"` **wie** `append` (anhängen) verwenden, wird neuer Inhalt immer nach dem bisherigen Inhalt platziert, sodass nichts verlorenggeht. Hier kann man den Unterschied gut sehen, wenn man beide Code-Varianten ausführt und am Ende die beiden Dateien vergleicht:

```
In [ ]: filename = "secrets_write.txt"

# Modus w
with open(filename, "w") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Python mehr als Java."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")

# Modus w
with open(filename, "w") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Kuhmilch mehr als Sojamilch."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")

# Modus w
with open(filename, "w") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Herbst lieber als Frühling."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")
```

```
In [ ]: filename = "secrets_append.txt"

# Modus a
with open(filename, "a") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Python mehr als Java."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")

# Modus a
with open(filename, "a") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Kuhmilch mehr als Sojamilch."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")

# Modus a
```

```
with open(filename, "a") as outfile:
    print("Secret:", file=outfile)
    content = "Ich mag Herbst lieber als Frühling."
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")
```

## Encodings

Im Beispiel oben kam ein Umlaut vor, der möglicherweise in den erzeugten Dateien nicht richtig dargestellt wurde. Das liegt daran, dass beim Schreiben von Dateien mehrere mögliche technische Umsetzungen der einzelnen Zeichen zur Verfügung stehen.

Um sicherzugehen, dass eine Datei korrekt geschrieben wird, können wir beim Öffnen das Encoding `utf8` als Parameter übergeben. Dann sieht unser Beispiel so aus:

```
In [ ]: filename = "secrets_umlaut.txt"

# Modus w
with open(filename, "w", encoding="utf8") as outfile:
    print("Secret:", file=outfile)
    content = "Schöne, ästhetische Umlaute für alle!!"
    print(content, file=outfile)
    print("Daten wurden in der Datei " + filename + " gespeichert!")
```

## Zusammenfassung

- Wir können Informationen in Dateien speichern, um sie dauerhaft verfügbar zu machen.
- Dateien können mit `with open(...) as outfile:` geöffnet werden.
- Wir können nur innerhalb des `with open` -Blocks auf die Datei zugreifen.
- Mit `print(..., file=outfile)` können wir Informationen in die Datei schreiben.
- Wir können Dateien im `w` -Modus komplett überschreiben, oder im `a` -Modus neue Inhalte anhängen, ohne die vorhandenen Inhalte zu löschen.
- Mit dem Parameter `encoding="utf8"` stellen wir sicher, dass auch Umlaute und andere Zeichen ordnungsgemäß dargestellt werden.

## Weitere Themen dieser Woche

- 04-01: Dictionarys
- 04-03: Dateien lesen